# PKCS #5:
# Password-Based Encryption Standard

*RSA Data Security, Inc.*
*Redwood City, CA*

*June 3, 1991*

*Version 1.4*[1]

## 1. Scope

This standard describes a method for encrypting an octet string with a secret key derived from a password. The result of the method is an octet string. Although the standard can be used to encrypt arbitrary octet strings, its intended primary application to public-key cryptography is for encrypting private keys when transferring them from one computer system to another, as described in PKCS #8 [1].

The standard defines two key-encryption algorithms: "MD2 with DES-CBC" and "MD5 with DES-CBC." The algorithms employ DES secret-key encryption [2] in cipher-block chaining mode [3], where the secret key is derived from a password with the MD2 message-digest algorithm [4,5], or MD5 message-digest algorithm [6].

## 2. References

[1] RSA Data Security, Inc. *PKCS #8: Private-Key Information Syntax Standard.* Version 1.1, June 1991.

[2] National Bureau of Standards. *FIPS Publication 46-1: Data Encryption Standard.* January 1988.

[3] National Bureau of Standards. *FIPS Publication 81: DES Modes of Operation.* December 1980.

[4] J. Linn. *RFC 1115: Privacy Enhancement for Internet Electronic Mail: Part III -- Algorithms, Modes, and Identifiers.* August 1989.

[5] B.S. Kaliski Jr. *RFC [MD2]: The MD2 Message-Digest Algorithm.* Draft, April 1991.

---

[1] Versions 1.0–1.3 are superseded by this version, The earlier versions were distributed to participants in RSA Data Security, Inc.'s Public-Key Cryptography Standards meetings in February and March 1991.

[6] Ronald L. Rivest and Stephen R. Dusse. *RFC [MD5]: The MD5 Message-Digest Algorithm.* In preparation.

[7] CCITT. *Recommendation X.509: The Directory—Authentication Framework.* 1988.

[8] CCITT. *Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1).* 1988.

[9] Robert Morris and Ken Thompson. Password security: A case history. *Communications of the ACM,* 22(11):594–597, November 1979.

## 3. Definitions

For the purposes of this standard, the following definitions apply.

`AlgorithmIdentifier`: A type that identifies an algorithm (by object identifier) and any associated parameters, as defined in X.509 [7].

**ASN.1:** Abstract Syntax Notation One, as defined in X.208 [8].

## 4. Symbols and abbreviations

Upper-case italic symbols (e.g., *C*) denote octet strings; lower-case italic symbols (e.g., *c*) denote integers.

`ab` hexadecimal octet value

*C* ciphertext

*EB* encryption block

*IV* initializing vector

*K* DES key

*M* message

*P* password

*PS* padding string

*S* salt value

*c* iteration count

*X* || *Y* concatenation of octet strings *X* and *Y*

$||X||$ length in octets of octet string $X$

mod $n$ arithmetic computation modulo $n$

## 5. General overview

The next three sections specify the encryption process, the decryption process, and object identifiers.

Each entity shall privately select an octet string $P$ as its "password." The password is an arbitrary octet string, and it need not be a printable "word" in the usual sense. The octet string may contain zero or more octets.

Each entity shall also select an eight-octet string $S$ as its salt value and a positive integer $c$ as its iteration count. The salt value $S$ and the iteration count $c$ are parameters of the algorithm identifier for the password-based encryption algorithm.

An entity's password, salt value, and iteration count may be different for each message the entity encrypts.

Two password-based encryption algorithms are defined here. The first algorithm (informally, "MD2 with DES-CBC") combines the MD2 message-digest algorithm [4,5] with DES in cipher-block chaining mode, and the second (informally, "MD5 with DES-CBC") combines the MD5 message-digest algorithm [6] with DES in cipher-block chaining mode. The "selected" message-digest algorithm shall either be MD2 or MD5, depending on the password-based encryption algorithm.

The encryption and decryption processes shall both be performed with the password, salt value, and iteration count as a key for the password-based encryption algorithm. Both processes transform an octet string to another octet string. The processes are inverses of one another if they use the same key.

**Notes.**

1. Some security conditions on the choice of password may well be taken into account in order to deter standard attacks such as dictionary attacks. These security conditions fall outside the scope of this standard.

2. How passwords are entered by an entity (a user) is outside the scope of this standard. However, it is recommended in the interest of interoperability that when messages encrypted according to this standard are to be transferred from one computer system to another, the password should consist of printable ASCII characters (values 32 to 126 decimal inclusive). This recommendation may require that password-entry software support optional conversion from a local character set to ASCII.

3. The iteration count provides a method of varying the time to derive a DES key from the password, thereby making dictionary attacks more expensive.

4. The selection of a salt value independent from the password limits the efficiency of dictionary attacks directed collectively against many messages encrypted according to this standard, e.g., against a database of encrypted private keys. This concept is explained in Morris and Thompson's paper on passwords [9].

5. A convenient way to select the salt value *S* to set it to the first eight octets of the message digest of the octet string $P \| M$. This method works best when the message *M* is somewhat random (e.g., a private key), for then the salt value *S* reveals no significant information about *M* or *P*. This method is not recommended, however, if the message *M* is known to belong to a small message space (e.g., "Yes" or "No").

6. The iteration count and the method of selecting a salt value may be standardized in particular applications.

## 6. Encryption process

This section describes the password-based encryption process.

The encryption process consists of three steps: DES key generation, encryption-block formatting, and DES encryption. The input to the encryption process shall be an octet string *M*, the message; an octet string *P*, the password; an octet string *S*, the salt value; and an integer *c*, the iteration count. The output from the encryption process shall be an octet string *C*, the ciphertext.

**Note.** The encryption process does not provide an explicit integrity check to facilitate error detection should the encrypted data be corrupted in transmission. However, the structure of the encryption block guarantees that the probability that corruption is undetected is less than $2^{-8}$.

### 6.1 DES key generation

A DES key *K* and an initializing vector *IV* shall be generated from the password *P*, the salt value *S*, and the iteration count *c*. The key generation process shall involve the following steps:

1. The octet string $P \| S$ shall be digested with *c* iterations of the selected message-digest algorithm. "One iteration" of the message-digest algorithm

is just the ordinary message digest; "two iterations" is the message digest of the message digest; and so on.

2. The least significant bit of each of the first eight octets of the result of step 1 shall be changed if necessary to give the octet odd parity, as required by DES. The resulting eight octets shall become the DES key *K*.

3. The last eight octets of the result of step 1 shall become the initializing vector *IV*.

### 6.2 Encryption-block formatting

The message *M* and a padding string *PS* shall be formatted into an octet string *EB*, the encryption block.

$$EB = M \,||\, PS \tag{1}$$

The padding string *PS* shall consist of $8 - (||M|| \bmod 8)$ octets all having value $8 - (||M|| \bmod 8)$. (This makes the length of the encryption block *EB* a multiple of eight octets.) In other words, the encryption block *EB* satisfies one of the following statements:

$$EB = M \,||\, \texttt{01} \text{ — if } ||M|| \bmod 8 = 7$$
$$EB = M \,||\, \texttt{02 02} \text{ — if } ||M|| \bmod 8 = 6$$
$$\vdots$$
$$EB = M \,||\, \texttt{08 08 08 08 08 08 08 08} \text{ — if } ||M|| \bmod 8 = 0$$

**Notes.** The encryption block can be parsed unambiguously since every encryption block ends with a padding string and no padding string is a suffix of another.

### 6.3 DES encryption

The encryption block *EB* shall be encrypted under DES in cipher-block chaining mode with key *K* and initializing vector *IV*. The result of encryption shall be an octet string *C*, the ciphertext.

**Note.** The length of the ciphertext *C* is a multiple of eight octets.

## 7. Decryption process

This section describes the password-based decryption process.

The decryption process consists of three steps: DES key generation, DES decryption, and encryption-block parsing. The input to the decryption process shall be an octet string *C*, the ciphertext; an octet string *P*, the password; an octet string *S*, the salt value; and an integer *c*, the iteration count. The output from the decryption process shall an octet string *M*, the message.

For brevity, the decryption process is described in terms of the encryption process.

### 7.1 DES key generation

A DES key *K* and an initializing vector *IV* shall be generated from the password *P*, the salt value *S*, and the iteration count *c* as described for the encryption process.

### 7.2 DES decryption

The ciphertext *C* shall be decrypted under DES in cipher-block chaining mode with key *K* and initializing vector *IV*. The result of decryption shall be an octet string *EB*, the encryption block.

It is an error if the length of the ciphertext *C* is not a multiple of eight octets.

### 7.3 Encryption-block parsing

The encryption block *EB* shall be parsed into an octet string *M*, the message, and an octet string *PS*, the padding string, according to Equation (1).

It is an error if the encryption block cannot be parsed according to Equation (1) (see notes to Section 6.2).

## 8. Object identifiers

This standard defines three object identifiers: `pkcs-5`, `pbeWithMD2AndDES-CBC`, and `pbeWithMD5AndDES-CBC`.

### 8.1 pkcs-5

The object identifier `pkcs-5` identifies this standard.

```
pkcs-5 OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) 5 }
```

The pkcs-5 object identifier is intended to be used only in defining other object identifiers in this standard.

### 8.2 pbeWithMD2AndDES-CBC

The object identifier pbeWithMD2AndDES-CBC identifies the "MD2 with DES-CBC" password-based encryption and decryption processes defined in Sections 6 and 7.

```
pbeWithMD2AndDES-CBC OBJECT IDENTIFIER ::= { pkcs-5 1 }
```

The pbeWithMD2AndDES-CBC object identifier is intended to be used in the algorithm field of a value of type AlgorithmIdentifier [7]. The parameters field of that type, which has the algorithm-specific syntax ANY DEFINED BY algorithm, would have ASN.1 type PBEParameter for this algorithm.

```
PBEParameter ::= SEQUENCE {
  salt OCTET STRING SIZE(8),
  iterationCount INTEGER }
```

The fields of type PBEParameter have the following meanings:

- salt is the salt value *S*.
- iterationCount is the iteration count *c*.

### 8.3 pbeWithMD5AndDES-CBC

The object identifier pbeWithMD5AndDES-CBC identifies the "MD5 with DES-CBC" password-based encryption and decryption processes defined in Sections 6 and 7.

```
pbeWithMD5AndDES-CBC OBJECT IDENTIFIER ::= { pkcs-5 3 }
```

The pbeWithMD5AndDES-CBC object identifier is intended to be used in the algorithm field of a value of type AlgorithmIdentifier. The parameters field of that type, which has the algorithm-specific syntax ANY DEFINED BY algorithm, would have ASN.1 type PBEParameter for this algorithm, as it does for the pbeWithMD2AndDES-CBC object identifier.